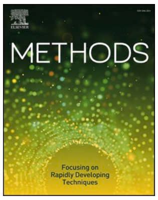
Contents lists available at ScienceDirect

Methods

journal homepage: www.elsevier.com/locate/ymeth



AGF-PPIS: A protein–protein interaction site predictor based on an attention mechanism and graph convolutional networks

Xiuhao Fu [a,1], Ye Yuan [b,1], Haoye Qiu [a], Haodong Suo [a], Yingying Song [a], Anqi Li [a], Yupeng Zhang [a], Cuilin Xiao [a], Yazi Li [a], Lijun Dou [c], Zilong Zhang [a,*], Feifei Cui [a,*]

[a] *School of Computer Science and Technology, Hainan University, Haikou 570228, China*
[b] *Beidahuang Industry Group General Hospital, Harbin 150001, China*
[c] *Genomic Medicine Institute, Lerner Research Institute, Cleveland, OH 44106, USA*

ARTICLE INFO

Keywords:
Protein–protein interactions
Multi-head self-attention
Graph convolutional network
Feed-forward neural network

ABSTRACT

Protein–protein interactions play an important role in various biological processes. Interaction among proteins has a wide range of applications. Therefore, the correct identification of protein–protein interactions sites is crucial. In this paper, we propose a novel predictor for protein–protein interactions sites, AGF-PPIS, where we utilize a multi-head self-attention mechanism (introducing a graph structure), graph convolutional network, and feed-forward neural network. We use the Euclidean distance between each protein residue to generate the corresponding protein graph as the input of AGF-PPIS. On the independent test dataset Test_60, AGF-PPIS achieves superior performance over comparative methods in terms of seven different evaluation metrics (ACC, precision, recall, F1-score, MCC, AUROC, AUPRC), which fully demonstrates the validity and superiority of the proposed AGF-PPIS model. The source codes and the steps for usage of AGF-PPIS are available at https://github.com/fxh1001/AGF-PPIS.

1. Introduction

Protein–protein interactions (PPIs) play crucial roles in cellular processes and determine the outcome of most cellular processes[1]. Understanding protein interactions can aid in the understanding of their molecular functions and their association with different biological processes[2–5]. PPIs are generally defined as physical contacts between proteins for molecular docking[6]. Identifying physical residues that touch each other in protein complexes helps to build protein–protein interaction networks[7], which contributes to elucidating the physiological mechanism of disease onset and thus can help to achieve effective diagnostic and therapeutic strategies[8]. The two most widely used methods for measuring PPIs are the binary and the co-complex methods [9], but they are time-consuming and expensive, and the process is relatively complicated[1]. Therefore, it is necessary to develop convenient PPIs site prediction methods.

At present, there are two main types of PPIs site prediction methods that are widely used: one is based on protein sequence, and the other is based on structure[10,11]. The prediction of sequence-based PPIs sites must only obtain the corresponding protein sequence. For example, the PPIs site predictor built by Sun et al. based on protein sequences using a stacked deep auto-encoder achieved satisfactory prediction results [12,13]. However, sequence-based predictors still have certain limitations[14–17]. In contrast, protein structure-based predictors, which incorporate the structural information of related proteins, can more accurately infer the PPIs sites[18,19].

The predictors of PPIs are often built based on machine learning technologies[20]. Many machine learning-based predictors are now successful[21,22]. For example, LightGBM[23], SVM[24–27], RF [28,29], XGBoost[30], and Naive Bayes[31] are used as predictive classifiers for PPIs sites. Of course, deep learning and has shown excellent performance in the prediction of PPIs sites[32]. For example, CNN [33,34], LSTM[35,36], GCN[37], AE[38] and other deep learning algorithms shine as classifiers for PPIs sites. However, to obtain better predictors of PPIs sites, effective feature representation of proteins is also essential[39]. Currently, feature representations extracted through protein sequence alignment generally contain evolutionary information, such as position-specific scoring matrices (PSSMs)[40–43] and hidden

[*] Corresponding authors.
E-mail addresses: zhangzilong@hainanu.edu.cn (Z. Zhang), feifeicui@hainanu.edu.cn (F. Cui).

[1] Equally contributed.

https://doi.org/10.1016/j.ymeth.2024.01.006
Received 21 September 2023; Received in revised form 4 January 2024; Accepted 13 January 2024
Available online 17 January 2024

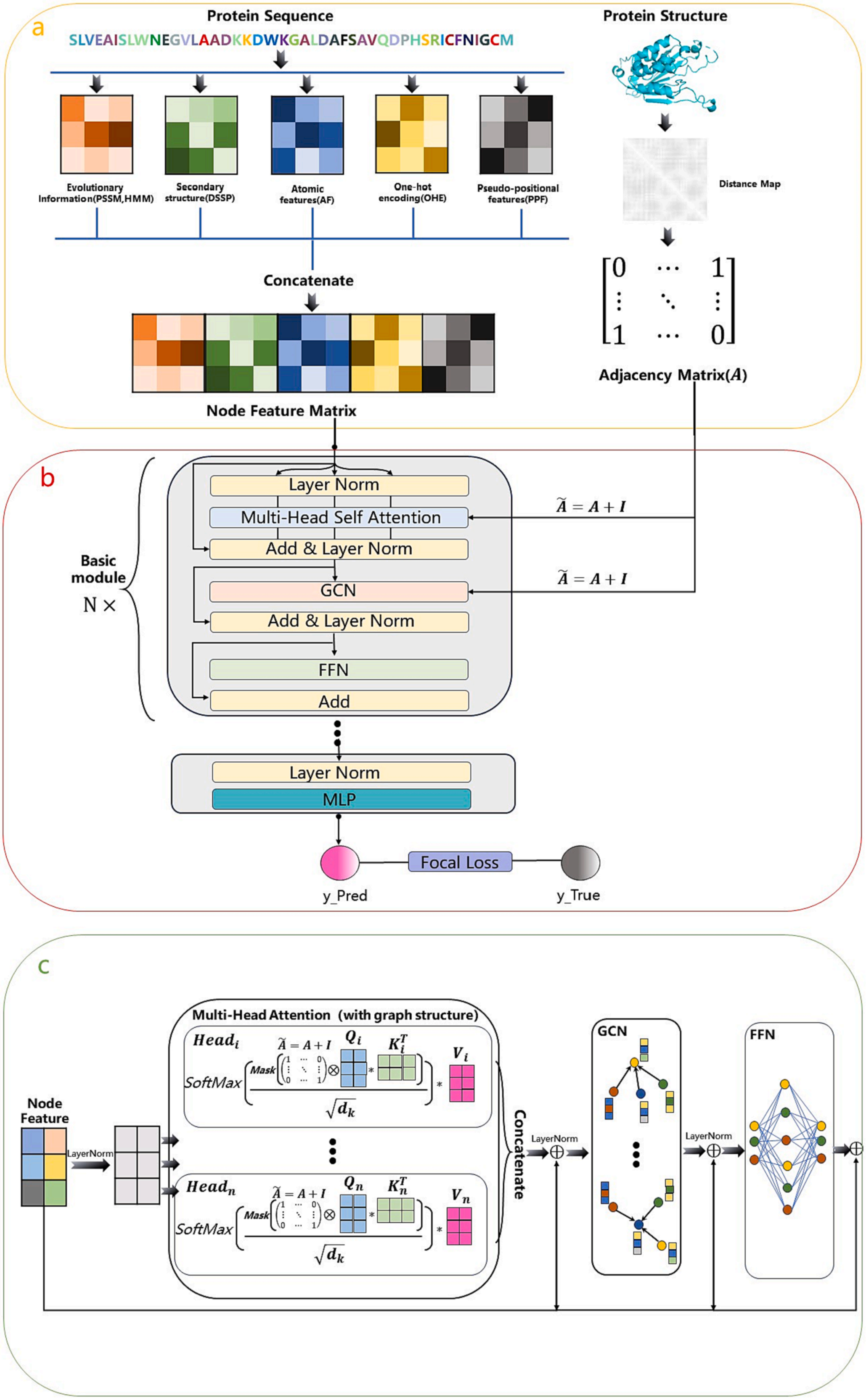


Fig. 1. The architecture of AGF-PPIS. Figure a represents feature extraction. Figure b represents the architecture of AGF-PPIS, where $\widetilde{A} = A + I$, A is the adjacency matrix, and I is the identity matrix. Figure c represents the detailed architecture of the basic modules that make up AGF-PPIS, where * represents matrix multiplication, ⊗ represents Hadamard product, ***Mask*** represents the mask operation, and ⊕ represents the summation operator.

Markov model (HMM) profiles[44,45]. Similarly, the feature representation of protein structure information is crucial for improving the prediction performance of PPIs sites, similar to the secondary structure of proteins[44,46]. There is also the use of protein physicochemical properties[47] and one hot encoding[48] of residues for feature representation. Based on a variety of feature representations, there are now many PPIs site predictors with good predictive effects. For example, Zeng et al. combined protein local context and global sequence features using a deep neural network to develop a predictor called DeepPPISP [48]. Li et al. used high-scoring segment pairs (HSPs), position information, and 3-mer amino acid embedding as three novel feature representation methods for the first time and used CNN and RNN to build an integrated model to develop the DELPHI predictor[49]. There is also the MaSIF-site[50] predictor developed by Gainza et al. In addition, there is the latest EDLMPPI[51] predictor developed by Hou et al., which extracts evolutionary information, physical and chemical properties and other features from protein sequences and uses a bidirectional long-term short-term memory network and capsule network to develop an ensemble deep memory capsule network. With these efficient PPIs site predictors, the prediction of PPIs sites becomes more convenient.

In the past few years, with the development of the application of graph convolutional networks (GCNs)[52], new ideas have been incorporated into the construction of PPIs site predictors. The residue chain of a protein can be regarded as a graph node, and the edges between each graph node are defined by the distance between each residue to construct the residue graph of the protein. GCN is used to capture the information carried by adjacent residues and to better predict PPIs by aggregating the information of adjacent residues[37]. Similarly, with the rise of large language models, such as typical Transformer[53] and BERT[54], attention mechanisms have begun to gain popularity[55]. Of course, we can add attention to improve the performance of the GCN. For example, GATs[56] constructed by introducing additive attention mechanisms have achieved new success in the prediction of PPIs sites [57].

In this study, we applied the multi-head self-attention (MSA) mechanism and GCN to build a predictor of PPIs sites (AGF-PPIS). For the one-layer architecture of the model, we borrowed the architecture of the transformer model and followed the MSA; however, due to the construction of the protein's graph structure, in order to better learn the information of adjacent nodes and remove redundant information from nonadjacent nodes, we introduced the residue graph structure of the protein in the calculation of the MSA, that is, by only calculating the MSA weight for a certain residue node and its adjacent residue nodes, and for nonadjacent residues, we did not participate in the calculation of attention weights. Then, GCN was used to further capture the feature information of adjacent nodes. Finally, the obtained results were further abstracted and learned using a feed-forward neural network (FFN) to deepen the understanding of the model. This single-layer architecture is stacked with multiple layers to obtain the final predictor AGF-PPIS. For protein feature representation, we extracted protein sequence evolution information, secondary structure information, atomic feature information, pseudo position information of residues, and one hot encoding of residues. At the same time, to alleviate the impact of uneven sample distribution, we used focal loss[58,59] as the loss function of AGF-PPIS for optimization. The architecture of the model is shown in Fig. 1.

2. Materials and methods

In this study, we transformed the prediction of PPIs sites into a graph node classification problem, extracted corresponding node features for each residue node, and finally determined the probability of whether each residue node is a PPIs site.

2.1. Dataset

The dataset used for this experiment is that used by the AGAT-PPIS [57] Institute, consisting of the training dataset (Train_335) and the test dataset (Test_60, Test_315 and UBtest_31). However, due to previous experiments, the sequences of some of the proteins are inconsistent with the sequences in the corresponding PDB files obtained on the PDB website. Therefore, we removed these inconsistent protein sequences from the dataset and finally obtained the training dataset (Train_334) and test dataset (Test_60, Test_280, UBtest_25). The above Train_335 and Test_60 are from widely used public datasets Dset_186, Dset_72[60] and Dset_164[61], which were filtered by six steps, and redundant proteins with high sequence similarity or overlap were removed by using BLASTClust[62]. Test_280 is another test dataset from GraphPPIS [37] research, with the purpose of further verifying the generalization ability of AGF-PPIS. UBtest_25 contains 25 unbound protein structures corresponding to 25 proteins in Test_60 that have known monomeric structures in PDB for evaluating the robustness of the model and the impact of the conformational changes on model performance. The specific statistical details of the datasets mentioned above are shown in Table 1.

Table 1
The specific statistical details of the datasets.

Dataset	Protein chains	Interacting residues	Noninteracting residues	of interacting residues (%)
Train_334	334	10,336	55,872	15.61
Test_60	60	2075	11,069	15.79
Test_280	280	8436	49,002	14.68
UBtest_25	25	711	5206	12.02

2.2. Protein representation

In our work, each protein is represented as an undirected graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{A})$, with $\boldsymbol{V}$ denoting residues, which is equivalent to the nodes in the graph, $\boldsymbol{E}$ denoting the contacts of residues according to pairwise distances, which is equivalent to the edges in the graph, and $\boldsymbol{A}$ is the adjacency matrix of graph $\boldsymbol{G}$.

2.2.1. Node features

To extract useful node features from protein sequences, we extracted amino acid evolution information (PSSM, HMM), structural properties (DSSP), atomic features, pseudo positional features, and one hot encoding[63]. These features were then concatenated to finally form the node feature matrix $\boldsymbol{X}_v \in \mathbb{R}^{n\times 88}$, where $\boldsymbol{n}$ denotes the number of residues in a protein and 88 indicates that we extracted an 88-dimensional node feature vector.

Evolutionary information. In the experiment, we used PSSM and HMM to represent the evolution information of residues. PSSM was produced by running PSI-BLAST v2.10.1[64]. The HMM was produced by running the HHblits v3.0.3[65] algorithm with default parameters. Each amino acid corresponds to a PSSM, and the HMM extracts 20-dimensional feature vectors. We then normalized the resulting PSSM and HMM matrices using Equation (1), thus constraining the values in both matrices to be between 0 and 1, where $\boldsymbol{v}$ is the input feature value, and $\boldsymbol{v}_{min}$ and $\boldsymbol{v}_{max}$ are the minimum and maximum values of a certain column of the input feature matrix.

$$v_{norm} = \frac{v - v_{min}}{v_{max} - v_{min}} \quad (1)$$

Secondary structure. For protein structural properties, we focused on the secondary structure of the protein[66]. First, a 9-dimensional one hot form feature vector was extracted from the secondary structure state. Second, the two torsion angles PHI and PSI of the polypeptide backbone were subjected to sin-cosine transformation to obtain a 4-dimensional feature vector. Third, the solvent-accessible surface area was converted into relative solvent accessibility to obtain a one-dimensional feature vector. Finally, we obtained 14-dimensional feature vectors for the structure of the protein. These structural feature vectors of proteins can be calculated by the DSSP algorithm[67], so this group of feature vectors was named DSSP.

Atomic features (AF). We considered adding features regarding the individual atoms (excluding hydrogen atoms) that make up the residue into the node feature matrix $\boldsymbol{X_v}$. Thus, seven features of atoms were extracted from the protein PDB file: whether it is in a ring and the van der Waals radius of the atom, B-factor, whether it is a residue side-chain atom, the number of hydrogen atoms bonded to it, electronic charge, and atomic mass. Considering that the number of atoms constituting each residue is not the same, we took the average of the seven atomic features in the residue as the final atomic feature, as shown in Equation (2). where $\{\boldsymbol{E_{i,j}}\}_{i=1,\cdots,7;j=1,\cdots N_A}$ denotes feature i of atom j in the residue, N_A denotes the number of atoms in a residue and $\{f_i\}_{i=1,\cdots,7}$ denotes atomic feature i of the residue. After processing, seven-dimensional atomic features were obtained for each residue.

$$f_i = \frac{1}{N_A}\sum_{j=1}^{j=N_A}\boldsymbol{E_{i,j}} \tag{2}$$

Pseudo positional features (PPF). We used the centroid coordinates of the side chains of the residues as the pseudo position information of the residues and then calculated the Euclidean distance between the pseudo position of each residue and the pseudo position of the reference residues. Finally, the obtained Euclidean distance value was divided by a hyperparameter λ to obtain a one-dimensional feature vector of each residue's pseudo position information. We took the first residue of the protein as the reference residue and its pseudo position as the reference $\boldsymbol{P_{ref}}$. The calculation step of the pseudo position feature $\boldsymbol{PPF_i}$ of node i is shown in Equation (3), where $\boldsymbol{P_i}$ denotes the pseudo position information of node i, $\boldsymbol{Euc}(\boldsymbol{P_i},\boldsymbol{P_{ref}})$ denotes the Euclidean distance between the pseudo position of node i and the reference position $\boldsymbol{P_{ref}}$, and λ is a hyperparameter.

$$\boldsymbol{PPF_i} = \frac{1}{\lambda}\boldsymbol{Euc}\left(\boldsymbol{P_i},\boldsymbol{P_{ref}}\right) \tag{3}$$

One hot encoding (OHE). Since there are 26 English letters in total, we used a 26-dimensional one hot encoding vector to encode the amino acid type in the protein.

2.3. The architecture of AGF-PPIS

AGF-PPIS is stacked by multiple basic modules, each of which includes an MSA, a GCN and an FFN. Finally, a multilayer perceptron (MLP) was used to convert the model output into predictions of PPIs.

2.3.1. Basic module

The basic module consists of an MSA, a GCN and an FFN, and normalization and activation functions are added between layers to make the model update the gradient more sufficiently and to prevent gradient explosion or gradient disappearance.

MSA layer. We applied the MSA mechanism learned from the transformer model to the protein graph and introduced the adjacency matrix of the corresponding protein graph when calculating the attention weight. The calculation process is as follows:

$$\boldsymbol{Q_i} = \boldsymbol{X_{in}W_i^Q}, \boldsymbol{K_i} = \boldsymbol{X_{in}W_i^K}, \boldsymbol{V_i} = \boldsymbol{X_{in}W_i^V}, \boldsymbol{i} = 1,\cdots,\boldsymbol{m} \tag{4}$$

$$\boldsymbol{Z_i} = \boldsymbol{Attention}\left(\boldsymbol{Q_i},\boldsymbol{K_i},\boldsymbol{V_i},\widetilde{\boldsymbol{A}}\right)_i$$

$$= \boldsymbol{Softmax}\left(\frac{\boldsymbol{Mask}\left(\widetilde{\boldsymbol{A}}\otimes(\boldsymbol{Q_iK_i^T})\right)}{\sqrt{\boldsymbol{d_k}}}\right)\boldsymbol{V_i}, \boldsymbol{i} = 1,\cdots,\boldsymbol{m}$$

$$\boldsymbol{Multi-Head}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = [\boldsymbol{Z_i}||..||\boldsymbol{Z_m}\ ]\boldsymbol{W_{cat}} \tag{5}$$

$$\boldsymbol{X_{out}} = \boldsymbol{X_{in}} + \boldsymbol{Multi-Head}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) \tag{6}$$

where $\boldsymbol{X_{in}}$ represents the input of the MSA, $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ are obtained by linear transformations of $\boldsymbol{X_{in}}$, and $\boldsymbol{m}$ represents the number of heads of MSA. Among them, $\widetilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$, $\boldsymbol{A}$ is the adjacency matrix of the protein graph, and $\boldsymbol{I}$ is the identity matrix. $\otimes$ indicates the Hadamard product. $\boldsymbol{Mask}$ represents the mask operation; that is, the attention weight of nonadjacent nodes is returned to 0. $\boldsymbol{Z_i}$ represents the attention obtained by each attention head. | | represents the concatenation operation. $\boldsymbol{X_{out}}$ represents the output of the MSA, and the result is obtained using the fitting residual.

GCN layer. We then used the GCN to further aggregate the features of adjacent nodes so that the model can learn more useful information from adjacent nodes. The formula of the GCN is as follows:

$$\boldsymbol{Z_{Gcn}} = \widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{X_{in}W_{Gcn}} \tag{7}$$

$$\boldsymbol{X_{out}} = \boldsymbol{X_{in}} + \boldsymbol{Z_{Gcn}} \tag{8}$$

where $\boldsymbol{X_{in}}$ denotes the input of the GCN, $\widetilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$, $\boldsymbol{A}$ is the adjacency matrix of the protein graph, $\boldsymbol{I}$ is the identity matrix, and $\widetilde{\boldsymbol{D}}$ is the degree matrix of $\widetilde{\boldsymbol{A}}$. $\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}$ normalizes the adjacency matrix $\widetilde{\boldsymbol{A}}$ (joining the self-loop). $\boldsymbol{W_{Gcn}}$ is the convolution kernel of the GCN, which is a trainable weight matrix. $\boldsymbol{Z_{Gcn}}$ is the result obtained after GCN. $\boldsymbol{X_{out}}$ is the output of the GCN, and the result is also obtained using the fitting residual.

FFN layer. Finally, we used the FFN to further abstract and elucidate the obtained results to deepen the understanding of the AGF-PPIS. The formula of FFN is as follows:

$$\boldsymbol{X_{out}} = \boldsymbol{X_{in}} + \boldsymbol{\sigma}\left(\boldsymbol{X_{in}W_1^{FFN}}\right)\boldsymbol{W_2^{FNN}} \tag{9}$$

where $\boldsymbol{X_{in}}$ denotes the input of the FFN. $\boldsymbol{\sigma}$ represents the ReLU activation function. $\boldsymbol{W_1^{FFN}}$ and $\boldsymbol{W_2^{FNN}}$ represent two debiased linear layer weight matrices, which perform two linear transformations on the input $\boldsymbol{X_{in}}$. $\boldsymbol{X_{out}}$ is the output of the FFN, and the result is still obtained using the fitting residual.

Normalization. In the basic module, we added a normalization operation before the MSA, GCN, and FFN. The normalization operation is as follows:

$$\boldsymbol{Norm}(\boldsymbol{x_i}) = \frac{\boldsymbol{x_i} - \boldsymbol{\mu}}{\sqrt{\boldsymbol{\sigma}^2 - \in}} \tag{10}$$

where $\boldsymbol{x_i}$ represents a column in the node feature matrix $\boldsymbol{X_v}$. $\boldsymbol{\mu}$ is the mean of $\boldsymbol{x_i}$, and $\boldsymbol{\sigma}^2$ is the variance of $\boldsymbol{x_i}$. Set the value of $\in$ to the negative 6th power of ten to prevent the denominator from being zero.

2.3.2. MLP

We passed the model output to the MLP[68,69] layer to obtain the probability of whether each residue is a PPIs site. Its calculation formula is as follows:

$$\boldsymbol{Y_{prob}} = \boldsymbol{Softmax}\left(\boldsymbol{X_{out}W_{mlp}} + \boldsymbol{b}\right) \tag{11}$$

where $\boldsymbol{X_{out}}$ represents the final output of the AGF-PPIS, $\boldsymbol{W_{mlp}}$ is the weight of the linear layer in the MLP, and $\boldsymbol{b}$ is the bias term. $\boldsymbol{Y_{prob}} \in \mathbb{R}^{n\times 2}$, ***where n*** is the number of residues in a protein. $\boldsymbol{Y_{prob}}$ has two columns, which are the predicted probability of being a PPIs site and not a PPIs site.

2.3.3. Focal loss

From Table 1, the distribution of samples in the dataset is very unbalanced. To alleviate the impact of this imbalance, we used focal loss as the loss function. Focal loss alleviates the sample distribution imbalance by assigning higher weights to difficult-to-learn samples and lower weights to easy-to-learn samples. The formula of focal loss is as follows:

$$\mathscr{L}_{FL} = \begin{cases} -(1-\alpha)(1-\widehat{p})^{\gamma}\boldsymbol{log}(\widehat{p})\boldsymbol{if}\,y = 1 \\ -\alpha\widehat{p}^{\gamma}\boldsymbol{log}(1-\widehat{p})\boldsymbol{if}\,y = 0 \end{cases} \tag{12}$$

where y represents the true label of the sample. $\widehat{p}$ is the probability that the model correctly predicts the sample label. α is the weight coefficient used to balance the influence of uneven sample distribution, which is a hyperparameter and participates in hyperparameter optimization. γ is also a hyperparameter, which is used to give smaller losses to samples that are easy to predict and greater losses to samples that are difficult to predict so that the model can focus on samples that are difficult to predict. Based on experience, we set it to 2.

2.4. Evaluation metrics

In the experiments, we used seven indicators, including the area under the precision-recall curve (AUPRC), area under the receiver operating characteristic curve (AUROC), accuracy (ACC), precision, recall, F1-score (F1) and Matthews-correlation coefficient (MCC), to evaluate our model. The calculation formulas of the evaluation metrics are as follows:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{16}$$

$$MCC = \frac{TP \times TN - FN \times FP}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{17}$$

where TP: true positive, TN: true negative, FP: false-positive and FN: false-negative. However, we only used AUPRC and AUROC for hyperparameter optimization because AUPRC and AUROC can fully reflect the model performance.

3. Experiment and discussions

In the experiment, we used AUROC and AUPRC as indicators to select hyperparameters. These two evaluation indicators can not only comprehensively evaluate the performance of the model but also do not need to select the threshold, so they save a certain amount of calculation and calculation time. The hyperparameters we selected are as follows. The adjacency matrix of the protein graph was generated by the Euclidean distance map between protein residues according to the distance threshold (the value is 1 when it is larger than or equal to the threshold value, and the value is 0 when it is smaller than the threshold value); this threshold value is 14 Å. For the pseudo position feature, the parameter λ was selected as 15 Å. In focal loss, the value of the weight α was selected as 0.9, and the value of γ was 2. For the model parameters, we chose the number of basic modules to be 12, and the number of neurons of a single hidden layer in the model is 256. Additionally, the number of neurons in the middle-hidden layer in the FFN was 516. The head number m of MSA was 6. For the learning rate, we used a decreasing mechanism, and the initial learning rate was 0.001. If the AUPRC indicator on the verification set did not increase in 10 iterations, we reduced the learning rate to 0.6 times and set the minimum learning rate to 10^{-6}. To prevent overfitting, we set the probability of dropout to 0.1. Finally, we set the maximum number of epochs during training to 80.

Table 2
The performance results of each feature ablation on the verification dataset and Test_60.

Feature	Validation set AUROC	Validation set AUPRC	Test_60 AUROC	Test_60 AUPRC
PSSM (EI)	0.835	0.508	0.849	0.538
HMM (EI)	0.850	0.560	0.855	0.557
DSSP (SS)	0.832	0.55	0.823	0.505
AF	0.798	0.418	0.806	0.472
PPF	0.847	0.549	0.856	0.553
OHE	0.860	0.570	0.849	0.570
ALL	**0.865**	**0.588**	**0.870**	**0.599**

3.1. Ablation experiment

We conducted ablation experiments on features and model architectures separately to evaluate the contribution of each feature and each model architecture.

3.1.1. Feature ablation

According to the hyperparameter settings mentioned above, we first conducted feature ablation experiments. In the experiment, we used five features of proteins: evolutionary information (PSSM, HMM), secondary structure (DSSP), AF, PPF, and OHE. Fivefold cross-validation is used to train and ablate these five features in turn. The performance results of each ablation on the verification dataset and the independent test dataset Test_60 are shown in Table 2. Here, we used two indicators, AUROC and AUPRC, for evaluation. For the values of AUROC and AUPRC, we chose the maximum value of the model on the validation datasets and Test_60 instead of the average value. From Table 2, when any feature is removed, the predictive power of the trained model on the verification dataset and the independent test dataset Test_60 is reduced, which shows that each feature plays a certain role and that there is no feature redundancy. Similarly, we also found that when the AF was removed, the AUROC value of the model on the validation dataset was 0.798, the AUPRC value was 0.418, the AUROC value on Test_60 was 0.806, and the AUPRC value was 0.472. The model achieved the lowest values of both AUROC and AUOPRC on the validation dataset and Test_60. We can see that the AF has an enormous impact on the learning ability of the model and gives more critical suggestions for the decision-making of the model. In contrast, we can also see that the model trained when removing the OHE feature, compared with other models for feature ablation, has the largest AUROC value and AUPRC value on the verification dataset, which are 0.860 and 0.570, respectively. On Test_60, the AUPRC value reaches the maximum value of 0.570. We can see that the OHE feature has the least influence on the performance of the model and is less important than other features.

Note: **ALL** (AGF-PPIS) represents the use of all features, that is, the performance of AGF-PPIS. Bold fonts indicate the best results, which is true in the following tables.

To evaluate the importance of features more comprehensively, we counted the values of AUROC and AUPRC of the optimal model for each fold in the 5-fold cross-validation in the feature ablation experiment on the validation dataset and the independent test dataset Test_60. Then, we added the AUROC and AUPRC of each fold and divided it by 2 (that is, average it) as the comprehensive evaluation indicator (AVG_Metric) of each fold. We drew radar charts of the AVG_Metric value on the validation dataset and Test_60, as shown in Fig. 2. To clearly represent the changes in AVG_Metric in each fold of cross-validation, we performed normalization. From Fig. 2, after removing the AF, AVG_Metric is the smallest on the verification data set in each fold cross-validation. On Test_60, after removing the AF, although the value of the cross-validation AVG_Metric per fold is not the smallest as on the

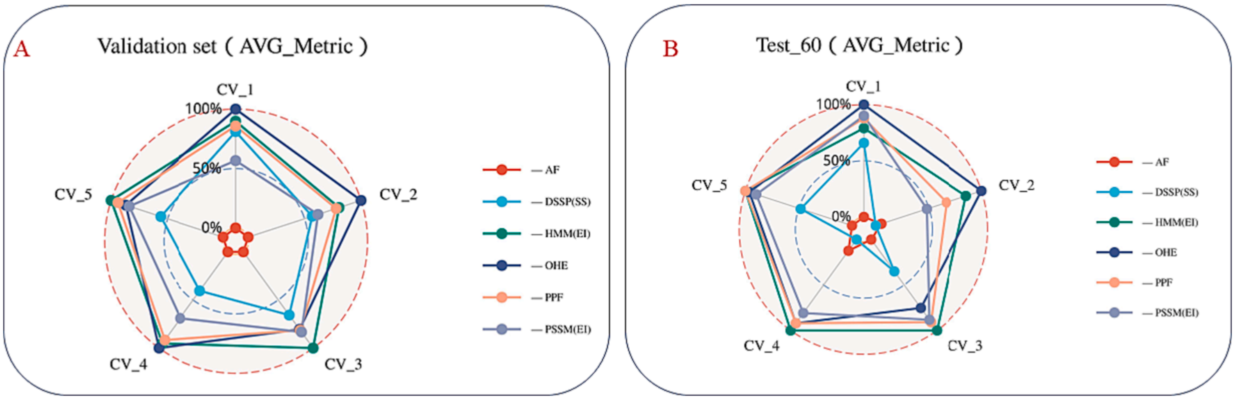


Fig. 2. The 5-fold cross-validation on the AVG_Metric value on the validation dataset (A) and Test_60 (B), where AVG_Metric = $\frac{AUROC+AUPRC}{2}$.

Table 3
The predictive power of the model trained using two OHE methods on Test_60.

	Validation set AUROC	Validation set AUPRC	Test_60 AUROC	Test_60 AUPRC
OHE (20)	0.852	0.540	0.854	0.567
OHE (26)	**0.865**	**0.588**	**0.870**	**0.599**

Table 4
The performance results of each model ablation on the verification dataset and Test_60.

Model ablation	Validation set AUROC	Validation set AUPRC	Test_60 AUROC	Test_60 AUPRC
GCN	0.844	0.545	0.855	0.544
FFN	0.848	0.557	0.854	0.548
Attn	0.864	0.566	0.854	0.56
ALL	**0.865**	**0.588**	**0.870**	**0.599**

verification dataset, the area enclosed by the AVG_Metric value obtained by each fold cross-validation is the smallest. Through Table 2 and Fig. 2 obtained from the ablation experiments of each feature, we found that the feature importance from high to low is AF > secondary structure (DSSP) > PSSM (evolutionary information) > PPF > HMM (evolutionary information) > OHE.

Similarly, for OHE features, we created 26-dimensional feature vectors based on the number of English letters instead of using the main twenty amino acids that make up proteins to create 20-dimensional feature vectors as in most other articles. We focused on the comparison of the contribution of these two OHE methods to the performance of this experimental model. When all parameters and feature encodings are the same, the performance of the model trained using two OHE methods on validation dataset and Test_60 is shown in Table 3. We also used AUROC and AUPRC for evaluation. From Table 3, whether on the validation dataset or Test_60, the 26-dimensional encoding method shows better AUROC and AUPRC than the 20-dimensional encoding method, so in this experiment, we used the 26-dimensional encoding method to extract the OHE feature of the protein.

Note: OHE (20) represents the performance of a model that uses 20 amino acids to extract features, and OHE (26) is the OHE method used by AGF-PIPIS.

3.1.2. Model ablation

The model architecture used in this experiment is shown in Chapter 2.3. The model we proposed (AGF_PPIS) is stacked by multiple basic modules. However, the basic modules include three basic architectures: the MSA that introduces the graph structure, the GCN and the FFN. These three basic architectures are the key to forming AGF_PPIS, so we first conduct ablation experiments on these three architectures. We also used two indicators, AUROC and AUPRC, for evaluation. The three basic architectures were removed from the model in turn. The predictive power of the trained models on the validation dataset and Test_60 is shown in Table 4. From Table 4, we can see that every time an architecture is eliminated, the predictive power of the model will decrease on the validation dataset and Test_60, which shows that each architecture is crucial. We can also see that when the GCN architecture is eliminated, the model's AUROC on the validation dataset dropped by 2.1 % and the AUPRC dropped by 4.3 %. On Test_60, the model's AUROC dropped by

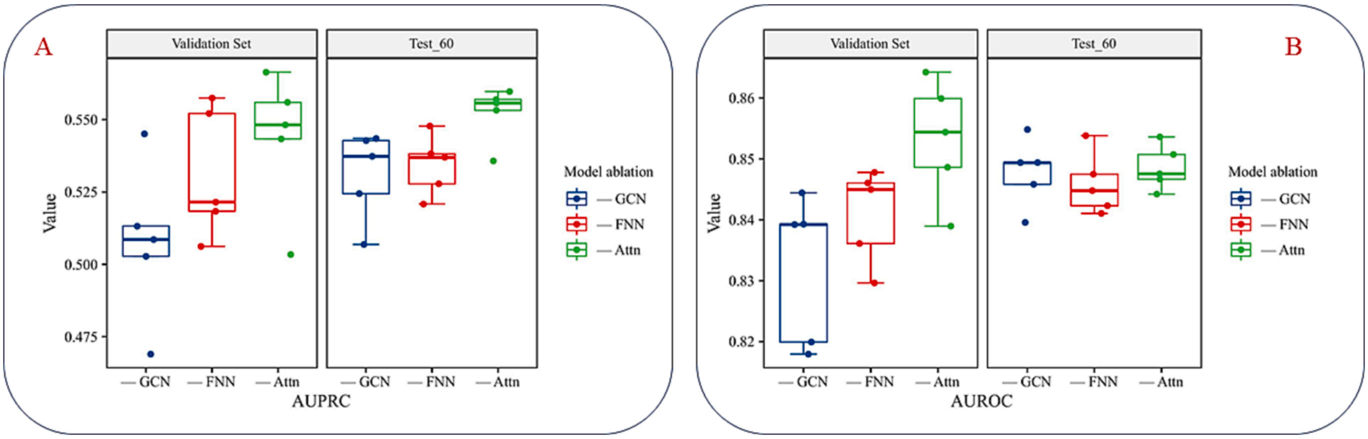


Fig. 3. The AUROC (A) and AUPRC (B) values of the optimal model at each fold in the 5-fold cross-validation on the validation dataset and Test_60.

Table 5
Comparison of the predictive power of the model trained with the Transformer's global attention mechanism and AGF-PPIS on Test_60.

	Validation set AUROC	Validation set AUPRC	Test_60 AUROC	Test_60 AUPRC
Attn (Global)-Model	0.858	0.561	0.854	0.567
AGF-PPIS	**0.865**	**0.588**	**0.870**	**0.599**

1.5 %, and the AUPRC dropped by 5.5 %. After eliminating the FFN, the model's AUROC (-1.7 %) and AUPRC (-3.1 %) on the validation dataset and AUROC (-1.6 %) and AUPRC (-5.1) on Test_60. After removing MSA, the model's AUROC (-0.1 %) and AUPRC (-2.2 %) on the validation dataset and AUROC (-1.6 %) and AUPRC (-3.9 %) on Test_60. From the results, we can see that when the GCN is eliminated, the predictive power of the model is greatly affected, regardless of whether it has the worst comprehensive performance on the verification dataset or Test_60. It can be concluded that the GCN is the most critical architecture of AGF-PPIS. In contrast, we can also see that the MSA has the least impact on the predictive power of the model.

Note that **ALL** (AGF-PPIS) represents the use of all architectures.

To comprehensively evaluate the importance of each architecture in AGF-PPIS, as in the feature ablation experiment, we counted the AUROC and AUPRC values of the optimal model at each fold in the 5-fold cross-validation and drew a box plot, as shown in Fig. 3. From Fig. 3, we can see more clearly that for the indicator AUPRC, when the GCN is removed, on the validation dataset, the model has the lowest average value of AUPRC and the lowest outlier. On Test_60, the AUPRC mean values between the models without GCN and the models without FFN are similar, but the model without GCN has the lowest outliers. Similarly, it can be clearly seen that, whether on the validation dataset or Test_60, the model that removes the MSA has the highest average AUPRC value in the 5-fold cross-validation. For the results for the AUROC indicator, on the validation dataset, similar to the AUPRC indicator, the model that removes the GCN has the smallest AUROC mean, and the model that removes the MSA has the largest AUROC mean. However, on Test_60, the model removing the FFN gives the lowest mean AUROC, the model removing the GCN gives the largest mean AUROC, and the model removing the MSA is in the middle. Although the model without the GCN gave the largest mean AUROC, it had the smallest outliers. Taken together, the GCN has the greatest impact on the model, followed by the FFN, and the MSA has the least impact on the model. However, we can see from Fig. 3 that each architecture plays an important role. For Test_60, the MSA also plays an important role.

The MSA mechanism used in this experiment introduces a graph structure, which is different from the MSA in Transformer. To verify that the introduced graph structure could enlarge the predictive power of the model, we also conducted a comparative experiment. Using the global MSA mechanism in the transformer, the other parameters and model architecture remained unchanged, and the model was retrained. The performance comparison with AGF-PPIS on validation dataset and Test_60 is shown in Table 5. Whether on the validation dataset or Test_60, the MSA mechanism introduced with the graph structure performs better in both AUROC and AUPRC, so the MSA mechanism with the graph structure is more beneficial to enlarge the predictive power of this experimental model (AGF-PPIS). At the same time, we chose a specific example, select a protein (PDB ID: 4E6N, chain: B), and we visualized the prediction results of the two models for this protein, as shown in Fig. 4. The false-positive (FP) samples predicted by the MSA mechanism model introduced into the graph structure (AGF-PPIS) are significantly reduced, and the predicted true positive (TP) samples also increased. We can see that the MSA mechanism introduced into the graph structure can indeed reduce the false-positive rate of prediction.

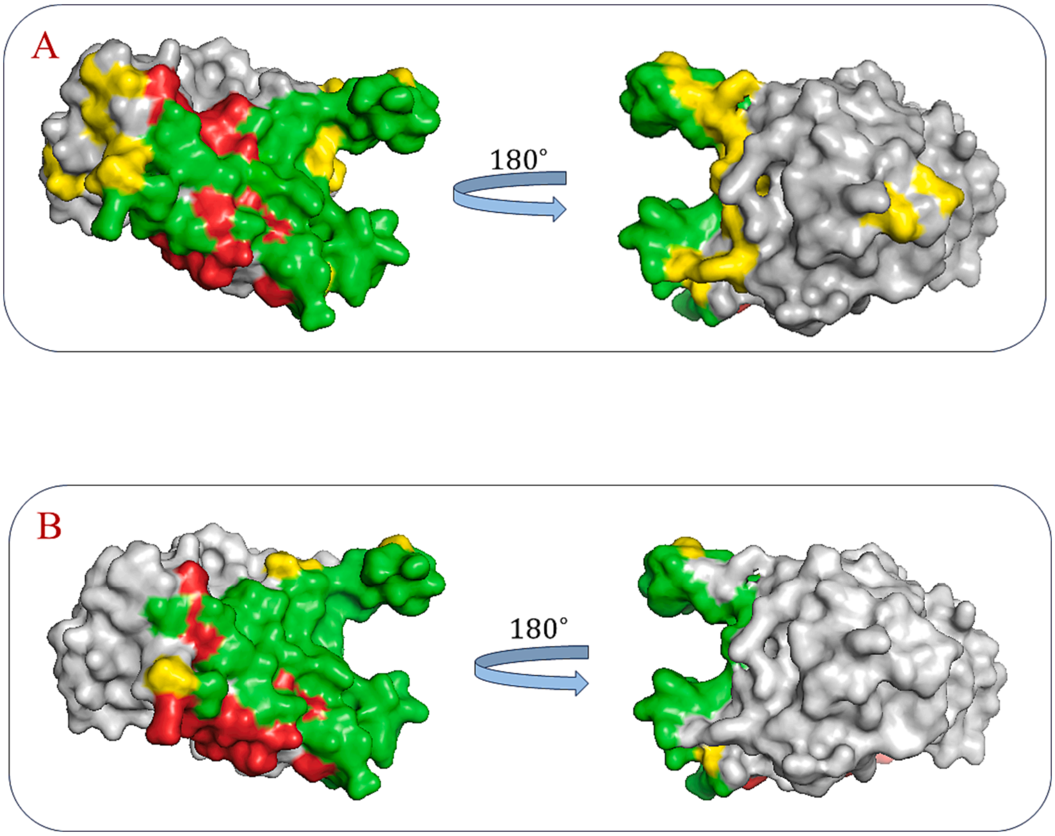


Fig. 4. (PDB ID: 4E6N, chain B) of PPIs site predictions by Attn (Global)-Model (A) and AGF-PPIS (B). Green: true positives, yellow: false-positives and red: false-negatives. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 6
Performance comparison of models trained with cross-entropy loss and models trained with focal loss (AGF-PPIS) on the validation dataset and Test_60.

	Validation set AUROC	Validation set AUPRC	Test_60 AUROC	Test_60 AUPRC
Cross-Entropy Loss	0.854	0.556	0.861	0.559
Focal Loss	**0.865**	**0.588**	**0.870**	**0.599**

To alleviate the influence of uneven sample distribution, we used focal loss as the loss function of model training. We also conducted comparative experiments. We used the cross-entropy loss function to retrain the model and compared it with the original model on the verification dataset and Test_60. The results are shown in Table 6. From Table 6, whether on the verification dataset or Test_60, focal loss makes the model perform better than the cross-entropy loss function, indicating that focal loss alleviates the impact of sample imbalance to a certain extent. Similarly, we selected a specific protein sample (PDB ID: 1QA9, chain: A) to visualize the prediction results of these two models trained with different loss functions, as shown in Fig. 5. The sequence length of this protein sample was 102, there were 19 PPIs sites, and the sample distribution was quite uneven. We focused on the predicted false-positive rate (FP). From Fig. 5b, the false-positive rate of the model using the cross-entropy loss is very high, covering almost all non-true positive (TP) residues. Compared with Fig. 5a, the model trained with focal loss has a dramatic reduction in the predicted false-positive rate. We can see that focal loss can indeed slow the impact of sample imbalance to a certain extent so that the model can learn in the right direction.

3.1.3. Predictive power comparison with other methods

We comprehensively compared AGF-PPIS with existing PPIs site predictors (DELPHI[49], DeepPPISP[48], SPPIDER[18], MaSIF-site[27], GraphPPIS[37], AGAT-PPIS[57]) on an independent test dataset Test_60, as shown in Table 7. Our AGF-PPIS model exceeds the existing optimal PPIs site predictors in all seven evaluation indicators of ACC (+0.4 %), precision (+1.2 %), recall (+1.7 %), F1-score (+1.5 %), MCC

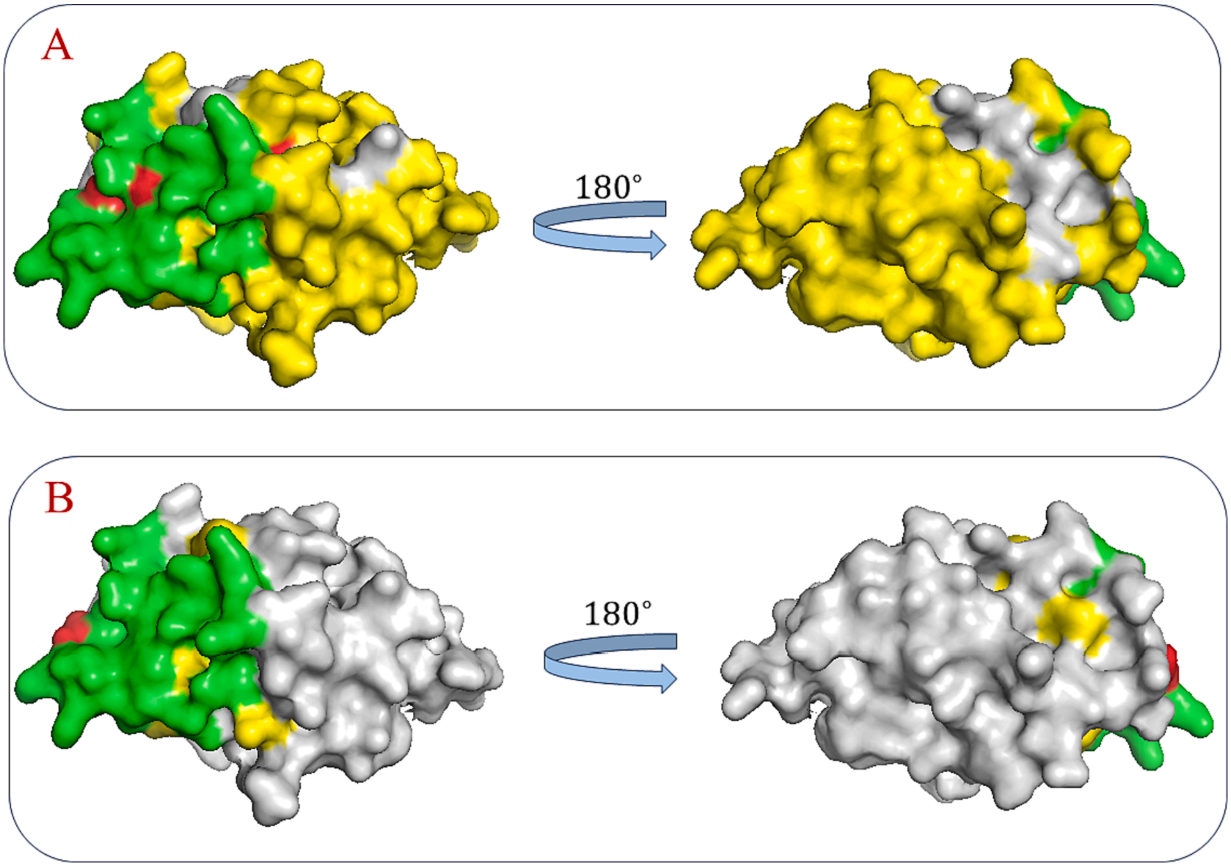


Fig. 5. (PDB ID:1QA9, chain A) of PPIs site predictions by the cross-entropy loss Model (A) and focal loss model (AGF-PPIS) (B). Green: true positives, yellow: false-positives and red: false-negatives. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 7
Comprehensively compared AGF-PPIS with existing PPIs site predictors on Test_60.

Predictor	ACC	Precision	Recall	F1-Score	MCC	AUROC	AUPRC
DELPHI[49]	0.697	0.276	0.568	0.372	0.225	0.699	0.319
DeepPPISP[48]	0.657	0.243	0.539	0.335	0.167	0.653	0.276
SPPIDER[18]	0.752	0.331	0.557	0.415	0.285	0.755	0.373
MaSIF-site[50]	0.780	0.370	0.561	0.446	0.326	0.775	0.439
GraphPPIS[37]	0.776	0.368	0.584	0.451	0.333	0.786	0.429
AGAT-PPIS[57]	0.856	0.539	0.603	0.569	0.484	0.867	0.574
AGF-PPIS	**0.860**	**0.551**	**0.620**	**0.584**	**0.501**	**0.870**	**0.599**

Table 8
Predictive power comparison of AGF-PPIS and other predictors on Test_280 and UBtest_25.

	Test_280		Btest_25		Ubtest_25	
	MCC	AUPRC	MCC	AUPRC	MCC	AUPRC
SPPIDER[18]	0.294	0.376	0.240	0.315	0.222	0.260
MaSIF-site[50]	0.304	0.372	0.217	0.299	0.141	0.225
GraphPPIS[37]	0.349	0.423	0.352	0.394	0.313	0.339
AGAT-PPIS[57]	0.481	**0.572**	0.485	0.583	0.327	0.365
AGF-PPIS	**0.484**	0.565	**0.518**	**0.604**	**0.339**	**0.370**

(1.7 %), AUROC (+0.3 %), and AUPRC (+2.5 %). To further evaluate the generalization ability and robustness of the model, we compared AGF-PPIS with other predictors on another test set, Test_280 and UBtest_25, and the results are shown in Table 8. We can see that on the test set Test_280, AGF-PPIS shows the best MCC, but AUPRC is slightly lower than the predictor AGAT-PPIS. However, AGF-PPIS performed optimally on both MCC and AUPRC on the UBtest_25 and Btest_25 datasets. Overall, AGF-PPIS is currently the model with the best predictive power. Of course, there is still room for improvement in its generalization ability.

Note: The results of other methods come from the paper AGAT-PPIS [33].

Note: Test_280 further verifies the generalization ability of the AGF-PPIS. Btest_31 represents the 31 bound structures in Test_60.

4. Conclusions

In this study, a novel site predictor of PPIs, AGF-PPIS, was proposed (https://github.com/fxh1001/AGF-PPIS). We constructed this site predictor using an MSA mechanism, GCN and FFN. We introduced the graph structure into the process of solving attention weights to better capture the association between nodes and their neighbors. At the same time, we added a FFN to strengthen the learning and understanding ability of the entire network. In the end, AGF-PPIS performed optimally on Test_60 and the test set UBtest_25, and AGF-PPIS is the current optimal model with strong robustness. However, according to the performance of the test set Test_280, the generalization ability of AGF-PPIS can be further improved. Similarly, based on algorithm improvements and sampling methods, the impact of sample imbalance can be further mitigated. Data enhancement techniques and adversarial training can also be introduced to further improve the robustness and generalization capabilities of AGF-PPIS. Of course, the structure of AGF-PPIS can also be used to predict the interaction sites of proteins and other ligands, such as peptides and drugs, thereby enhancing the versatility of AGF-PPIS.

The work was supported by the National Natural Science Foundation of China (No. 62101100, 62262015 and 62002242) and the National Natural Science Foundation of Sichuan Province of China (No. 2022NSFSC0943).

CRediT authorship contribution statement

Xiuhao Fu: Data curation, Writing – original draft, Software. **Ye Yuan:** Conceptualization, Methodology, Writing – review & editing. **Haoye Qiu:** Visualization, Investigation, Validation. **Haodong Suo:** Writing – review & editing. **Yingying Song:** Writing – review & editing. **Anqi Li:** Writing – review & editing. **Yupeng Zhang:** Writing – review & editing. **Cuilin Xiao:** Conceptualization. **Yazi Li:** Conceptualization. **Lijun Dou:** Conceptualization. **Zilong Zhang:** Supervision. **Feifei Cui:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

I have shared the link to my code and dataset at the end of the Manuscript File.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.ymeth.2024.01.006.

References

[1] Shoemaker BA, Panchenko AR. Deciphering Protein–Protein Interactions. Part I. Experimental Techniques and Databases, PLOS Computat. Biol. 2007;3:e42.

[2] N. Orii, M.K. Ganapathiraju, Wiki-Pi: A Web-Server of Annotated Human Protein-Protein Interactions to Aid in Discovery of Protein Function, PLoS One 7 (2012) e49029.

[3] F. Cui, Z. Zhang, C. Cao, et al., Protein-DNA/RNA interactions: Machine intelligence tools and approaches in the era of artificial intelligence and big data, Proteomics 22 (2022) e2100197.

[4] Wang Y, Zhai Y, Ding Y et al. SBSM-Pro: Support Bio-sequence Machine for Proteins, arXiv e-prints 2023:arXiv:2308.10275-arXiv:12308.10275.

[5] Y. Ding, J. Tang, F. Guo, Identification of Drug-Target Interactions via Dual Laplacian Regularized Least Squares with Multiple Kernel Fusion, Knowl.-Based Syst. 204 (2020) 106254.

[6] R.J. De Las, C. Fontanillo, Protein–protein interaction networks: unraveling the wiring of molecular machines within the cell, Brief. Funct. Genomics 11 (2012) 489–496.

[7] G. Alanis-Lobato, M.A. Andrade-Navarro, M.H. Schaefer, HIPPIE v2.0: enhancing meaningfulness and reliability of protein–protein interaction networks, Nucleic Acids Res. 45 (2016) D408–D414.

[8] N. Safari-Alighiarloo, M. Taghizadeh, M. Rezaei-Tavirani, et al., Protein-protein interaction networks (PPI) and complex diseases, Gastroenterol Hepatol. Bed Bench 7 (2014) 17–31.

[9] R.J. De Las, C. Fontanillo, Protein-Protein Interactions Essentials: Key Concepts to Building and Analyzing Interactome Networks, PLoS Comput. Biol. 6 (2010) e1000807.

[10] Wang Y, Zhai Y, Ding Y et al. SBSM-Pro: Support Bio-sequence Machine for Proteins, arXiv preprint arXiv:2308.10275 2023.

[11] R. Wang, Y. Jiang, J. Jin, et al., DeepBIO: an automated and interpretable deep-learning platform for high-throughput biological sequence prediction, functional annotation and visualization analysis, Nucleic Acids Res. 51 (2023) 3017–3029.

[12] T. Sun, B. Zhou, L. Lai, et al., Sequence-based prediction of protein protein interaction using a deep-learning algorithm, BMC Bioinf. 18 (2017) 277.

[13] Teng Z, Zhang Y, Dai Q et al. Constructing discriminative feature space for LncRNA-protein interaction based on deep autoencoder and marginal fisher analysis, Comput. Biol. Med. 2023;157.

[14] J. Zhang, L. Kurgan, Review and comparative assessment of sequence-based predictors of protein-binding residues, Brief. Bioinform. 19 (2017) 821–837.

[15] H. Shi, Y. Li, Y. Chen, et al., ToxMVA: An end-to-end multi-view deep autoencoder method for protein toxicity prediction, Comput. Biol. Med. 151 (2022).

[16] J. Jin, Y. Yu, R. Wang, et al., iDNA-ABF: multi-scale deep biological language learning model for the interpretable prediction of DNA methylations, Genome Biol. 23 (2022) 1–23.

[17] F. Cui, Z. Zhang, Q. Zou, Sequence representation approaches for sequence-based protein prediction tasks that use deep learning, Brief. Funct. Genomics 20 (2021) 61–73.

[18] A. Porollo, J. Meller, Prediction-based fingerprints of protein–protein interactions, Proteins: Structure, Function Bioinformat. 66 (2007) 630–645.

[19] L. Wei, W. Long, L. Wei, MDL-CPI: Multi-view deep learning model for compound-protein interaction prediction, Methods 204 (2022) 418–427.

[20] Z. Lv, M. Li, Y. Wang, et al., Editorial: Machine learning for biological sequence analysis, Front. Genet. 14 (2023) 1150688.

[21] D. Sarkar, S. Saha, Machine-learning techniques for the prediction of protein–protein interactions, J. Biosci. 44 (2019) 104.

[22] Z. Lv, C. Ao, Q. Zou, Protein Function Prediction: From Traditional Classifier to Deep Learning, Proteomics 19 (2019) e1900119.

[23] C. Chen, Q. Zhang, Q. Ma, et al., LightGBM-PPI: Predicting protein-protein interactions through LightGBM with multi-information fusion, Chemom. Intel. Lab. Syst. 191 (2019) 54–64.

[24] S. Romero-Molina, Y.B. Ruiz-Blanco, M. Harms, et al., PPI-Detect: A support vector machine model for sequence-based prediction of protein–protein interactions, J. Comput. Chem. 40 (2019) 1233–1242.

[25] P. Joshi, V. Masilamani, R. Ramesh, An Ensembled SVM Based Approach for Predicting Adverse Drug Reactions, Curr. Bioinform. 16 (2021) 422–432.

[26] H. Lin, C. Jian, Y. Cao, et al., MDD-TSVM: A novel semisupervised-based method for major depressive disorder detection using electroencephalogram signals, Comput. Biol. Med. 140 (2022).

[27] C. Ao, X. Ye, T. Sakurai, et al., m5U-SVM: identification of RNA 5-methyluridine modification sites based on multi-view features of physicochemical features and distributed representation, BMC Biol. 21 (2023).

[28] T.C. Northey, A. Barešić, A.C.R. Martin, IntPred: a structure-based predictor of protein–protein interaction sites, Bioinformatics 34 (2017) 223–229.

[29] S. Jiao, Q. Zou, H. Guo, et al., iTTCA-RF: a random forest predictor for tumor T cell antigens, J. Transl. Med. 19 (2021).

[30] C. Chen, Q. Zhang, B. Yu, et al., Improving protein-protein interactions prediction accuracy using XGBoost feature selection and stacked ensemble classifier, Comput. Biol. Med. 123 (2020) 103899.

[31] Q.C. Zhang, D. Petrey, L. Deng, et al., Structure-based prediction of protein–protein interactions on a genome-wide scale, Nature 490 (2012) 556–560.

[32] F. Soleymani, E. Paquet, H. Viktor, et al., Protein–protein interaction prediction with deep learning: A comprehensive review, Comput. Struct. Biotechnol. J. 20 (2022) 5316–5341.

[33] L. Wang, H.-F. Wang, S.-R. Liu, et al., Predicting Protein-Protein Interactions from Matrix-Based Protein Sequence Using Convolution Neural Network and Feature-Selective Rotation Forest, Sci. Rep. 9 (2019) 9848.

[34] Z. Ma, Y. Qi, C. Xu, et al., ATFE-Net: Axial Transformer and Feature Enhancement-based CNN for ultrasound breast mass segmentation, Comput. Biol. Med. 153 (2023).

[35] S. Tsukiyama, M.M. Hasan, S. Fujii, et al., LSTM-PHV: prediction of human-virus protein–protein interactions by LSTM with word2vec, Brief. Bioinform. 22 (2021).

[36] F. Cui, S. Li, Z. Zhang, et al., DeepMC-iNABP: Deep learning for multiclass identification and classification of nucleic acid-binding proteins, Comput. Struct. Biotechnol. J. 20 (2022) 2020–2028.

[37] Q. Yuan, J. Chen, H. Zhao, et al., Structure-aware protein–protein interaction site prediction using deep graph convolutional network, Bioinformatics 38 (2021) 125–132.

[38] Y.-B. Wang, Z.-H. You, X. Li, et al., Predicting protein–protein interactions from protein sequences by a stacked sparse autoencoder deep neural network, Mol. Biosyst. 13 (2017) 1336–1344.

[39] J. Zeng, D. Li, Y. Wu, et al., An empirical study of features fusion techniques for protein-protein interaction prediction, Curr. Bioinform. 11 (2016) 4–12.

[40] J. Zahiri, O. Yaghoubi, M. Mohammad-Noori, et al., PPIevo: Protein–protein interaction prediction from PSSM based evolutionary information, Genomics 102 (2013) 237–242.

[41] Q. Li, J. Yu, Y. Yan, et al., PsePSSM-based Prediction for the Protein-ATP Binding Sites, Curr. Bioinform. 16 (2021) 576–582.

[42] H.V. Tran, Q.H. Nguyen, iAnt: Combination of Convolutional Neural Network and Random Forest Models Using PSSM and BERT Features to Identify Antioxidant Proteins, Curr. Bioinform. 17 (2022) 184–195.

[43] C. Ao, L. Yu, Q. Zou, Prediction of bio-sequence modifications and the associations with diseases, Brief. Funct. Genomics 20 (2021) 1–18.

[44] Y. Xia, C.-Q. Xia, X. Pan, et al., GraphBind: protein structural context embedded rules learned by hierarchical graph neural networks for recognizing nucleic-acid-binding residues, Nucleic Acids Res. 49 (2021) e51–e.

[45] Y. Ding, W. He, J. Tang, et al., Laplacian Regularized Sparse Representation Based Classifier for Identifying DNA N4-Methylcytosine Sites via L(2,1/2)-Matrix Norm, IEEE/ACM Trans. Comput. Biol. Bioinf. 20 (2023) 500–511.

[46] S. Jiao, Q. Zou, Identification of plant vacuole proteins by exploiting deep representation learning features, Comput. Struct. Biotechnol. J. 20 (2022) 2921–2927.

[47] B. Zhang, J. Li, L. Quan, et al., Sequence-based prediction of protein-protein interaction sites by simplified long short-term memory network, Neurocomputing 357 (2019) 86–100.

[48] M. Zeng, F. Zhang, F.-X. Wu, et al., Protein–protein interaction site prediction through combining local and global features with deep neural networks, Bioinformatics 36 (2019) 1114–1120.

[49] Y. Li, G.B. Golding, L. Ilie, DELPHI: accurate deep ensemble model for protein interaction sites prediction, Bioinformatics 37 (2020) 896–904.

[50] P. Gainza, F. Sverrisson, F. Monti, et al., Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning, Nat. Methods 17 (2020) 184–192.

[51] Z. Hou, Y. Yang, Z. Ma, et al., Learning the protein language of proteome-wide protein-protein binding sites via explainable ensemble deep learning, Communications Biology 6 (2023) 73.

[52] Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks, arXiv preprint arXiv:1609.02907 2016.

[53] Vaswani A, Shazeer N, Parmar N et al. Attention is all you need, Advances in neural information processing systems 2017;30.

[54] Devlin J, Chang M-W, Lee K et al. Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 2018.

[55] Q. Jin, H. Cui, C. Sun, et al., Free-form tumor synthesis in computed tomography images via richer generative adversarial network, Knowl.-Based Syst. 218 (2021) 106753.

[56] Veličković P, Cucurull G, Casanova A et al. Graph attention networks, arXiv preprint arXiv:1710.10903 2017.

[57] Y. Zhou, Y. Jiang, Y. Yang, AGAT-PPIS: a novel protein–protein interaction site predictor based on augmented graph attention network with initial residual and identity mapping, Brief. Bioinform. 24 (2023).

[58] T.-Y. Lin, P. Goyal, R. Girshick, et al., Focal loss for dense object detection, in: In: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2980–2988.

[59] S. Das, U. Pradhan, N.S. Rai, Five Years of Gene Networks Modeling in Single-cell RNA-sequencing Studies: Current Approaches and Outstanding Challenges, Curr. Bioinform. 17 (2022) 888–908.

[60] Y. Murakami, K. Mizuguchi, Applying the Naïve Bayes classifier with kernel density estimation to the prediction of protein–protein interaction sites, Bioinformatics 26 (2010) 1841–1848.

[61] K. Dhole, G. Singh, P.P. Pai, et al., Sequence-based prediction of protein–protein interaction sites with L1-logreg classifier, J. Theor. Biol. 348 (2014) 47–54.

[62] S.F. Altschul, W. Gish, W. Miller, et al., Basic local alignment search tool, J. Mol. Biol. 215 (1990) 403–410.

[63] C. Cao, J. Wang, D. Kwok, et al., webTWAS: a resource for disease candidate susceptibility genes identified by transcriptome-wide association study, Nucleic Acids Res. 50 (2022) D1123–D1130.

[64] S.F. Altschul, T.L. Madden, A.A. Schäffer, et al., Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, Nucleic Acids Res. 25 (1997) 3389–3402.

[65] M. Remmert, A. Biegert, A. Hauser, et al., HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment, Nat. Methods 9 (2012) 173–175.

[66] A.K. Sharma, R. Srivastava, Protein Secondary Structure Prediction Using Character bi-gram Embedding and Bi-LSTM, Curr. Bioinform. 16 (2021) 333–338.

[67] Kabsch W, Sander C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features, Biopolymers: Original Research on Biomolecules 1983;22:2577-2637.

[68] S. Tang, L. Chen, iATC-NFMLP: Identifying Classes of Anatomical Therapeutic Chemicals Based on Drug Networks, Fingerprints, and Multilayer Perceptron, Curr. Bioinformat. 17 (2022) 814–824.

[69] O.C. Arican, O. Gumus, PredDRBP-MLP: Prediction of DNA-binding proteins and RNA-binding proteins by multilayer perceptron, Comput. Biol. Med. 164 (2023), 107317-107317.